

What is claimed is:

1. A method for processing an input document encoded in an extensible human-friendly extensible markup language ("XML"), said method comprising the steps of:

- 5 (a) converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML");
- (b) processing said output document encoded in mXML;
- (c) identifying a target to which the processed output document will be next routed; and
- 10 (d) determining whether said target is capable of processing documents encoded in mXML.

2. The method of claim 1, wherein step (c) comprises parsing the processed output document.

3. The method of claim 2, wherein step (c) comprises identifying a host name string for routing of the processed output document.

4. The method of claim 1, wherein step (d) comprises referencing a datastore, said  
20 datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.

5. The method of claim 1, wherein step (a) comprises the steps of:

- (a1) creating a document tree representation of the input document;
- (a2) obtaining a node count representing a count of nodes in the document tree representation;
- (a3) writing the node count to an mXML buffer;
- 5 (a4) traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising the steps of:
- generating a node name;
- generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;
- 10 generating a child list specifying index values of zero or more nodes which are children of the node; and
- generating a node value specification, which is empty if the node has no value;
- (a5) generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and
- 15 (a6) appending the data buffer to the mXML buffer to form the output document.
6. The method of claim 1, wherein step (b) comprises the step of:
- (b1) parsing said output document.
- 20
7. The method of claim 6, wherein step (b1) comprises the step of:
- (b1a) extracting routing data from said output document.

8. The method of claim 1, wherein step (b) comprises the step of:

(b1) transmitting the processed output document to said target if said target is determined in step (d) to be capable of processing documents encoded in mXML.

5 9. The method of claim 1, wherein step (b) comprises the step of:

(b1) processing said output document for content based routing if said target is determined in step (d) to be capable of processing documents encoded in mXML.

10. The method of claim 1, further comprising the step of:

(e) converting said processed output document encoded in mXML to an output document encoded in XML if said target is determined in step (d) to be not capable of processing documents encoded in mXML.

11. The method of claim 10, further comprising the step of:

(f) processing the converted output document encoded in XML.

12. The method of claim 11, wherein step (f) comprises transmitting the converted output document encoded in XML to said target.

13. The method of claim 6, wherein step (b1) comprises the steps of:

(b1a) parsing a node count representing a count of nodes in the document;

(b1b) parsing a node specification for each of the nodes, further comprising the steps of:

parsing a node name;

parsing a child list specifying index values of zero or more

nodes which are children of the node;

parsing an attribute list specifying zero or more (attribute

5 name, attribute value) pair references for attributes of the node; and

parsing a node value specification, which is empty if the node

has no value; and

(b1c) parsing a data buffer containing attribute names and attribute values

referenced from the attribute lists and node values referenced from the node value

10 specifications.

14. A method for processing an input document encoded in a machine-oriented extensible markup language ("mXML"), said method comprising the steps of:

(a) determining whether said input document will be next routed to a target

15 which is capable of processing documents encoded in mXML;

(b) converting said input document encoded in mXML to an output document

encoded in a human friendly extensible markup language ("XML") if said target is

determined in step (a) to be not capable of processing documents encoded in mXML; and

(c) processing said output document encoded in XML.

20 15. The method of claim 14, wherein step (a) comprises the steps of:

(a1) identifying a target to which said input document will be next routed; and

(a2) determining whether said target is capable of processing documents encoded

in mXML.

16. The method of claim 15, wherein step (a2) comprises referencing a datastore, said  
datastore storing data identifying a plurality of targets and indicating whether each of said  
5 plurality of targets is capable of processing documents encoded in mXML.

17. The method of claim 14, wherein step (b) comprises the steps of:

(b1) identifying a document tree representation of the input document;

(b2) reading a node count from an mXML buffer;

10 (b3) traversing each node in the document tree representation and generating a  
corresponding node specification in the mXML buffer, further comprising the steps of:

identifying a node name;

identifying an attribute list specifying zero or more (attribute name, attribute  
value) pair references for attributes of the node;

15 identifying a child list specifying index values of zero or more nodes which  
are children of the node; and

identifying a node value specification, which is empty if the node has no  
value;

(b4) generating a data buffer containing attribute names and attribute values  
20 referenced from the attribute lists and node values referenced from the node value  
specifications; and

(b5) appending the data buffer to the XML buffer to form the output document.

18. The method of claim 14, further comprising the step of:

(d) processing said input document encoded in mXML.

19. The method of claim 18, wherein step (d) comprises parsing said input document  
5 encoded in mXML.

20. The method of claim 14, wherein step (c) comprises transmitting said output  
documents encoded in XML.

21. A computer program product embodied on one or more computer-readable media, the  
10 computer program product adapted for processing an input document encoded in an  
extensible human-friendly extensible markup language ("XML") and comprising:

computer-readable program code for converting said input document encoded in  
XML to an output document encoded in a machine-oriented extensible markup language  
15 ("mXML");

computer-readable program code for processing said output document encoded in  
XML;

computer-readable program code for identifying a target to which the processed  
output document will be next routed; and

20 computer-readable program code for determining whether said target is capable of  
processing documents encoded in mXML.

22. The computer program product of claim 21, wherein said computer-readable program

code for converting said input document encoded in an extensible human-friendly extensible markup language (XML) to an output document encoded in a machine-oriented extensible markup language ("mXML") comprises:

computer-readable program code means for creating a document tree representation of the input document;

computer-readable program code means for obtaining a node count representing a count of nodes in the document tree representation;

computer-readable program code means for writing the node count to an mXML buffer;

computer-readable program code means for traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising:

computer-readable program code means for generating a node name;

computer-readable program code means for generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

computer-readable program code means for generating a child list specifying index values of zero or more nodes which are children of the node; and

computer-readable program code means for generating a node value specification, which is empty if the node has no value;

computer-readable program code means for generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

computer-readable program code means for appending the data buffer to the mXML buffer to form the output document.

23. The computer program product of claim 21, wherein said computer-readable program code for processing said output document comprises computer-readable program code for processing a document encoded in mXML comprising:

computer-readable program code means for parsing the document, further comprising:

computer-readable program code means for parsing a node count representing a count of nodes in the document;

computer-readable program code means for parsing a node specification for each of the nodes, further comprising:

computer-readable program code means for parsing a node name;

computer-readable program code means for parsing a child list

specifying index values of zero or more nodes which are children of the node;

computer-readable program code means for parsing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

computer-readable program code means for parsing a node value specification, which is empty if the node has no value; and

computer-readable program code means for parsing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

computer-readable program code means for using the parsed document as input for the processing.

24. A system for processing an input document encoded in an extensible human-friendly extensible markup language ("XML"), said system comprising:

means for converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML");

means for processing said output document encoded in mXML;

means for identifying a target to which the processed output document will be next routed; and

means for determining whether said target is capable of processing documents encoded in mXML.

25. The system of claim 24, wherein said means for converting said output document encoded in an extensible human-friendly extensible markup language (XML) to an output document encoded in a machine-oriented extensible markup language (mXML) comprises:

means for creating a document tree representation of the input document;

means for obtaining a node count representing a count of nodes in the document tree representation;

means for writing the node count to an mXML buffer;

means for traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising:

means for generating a node name;

means for generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

means for generating a child list specifying index values of zero or more nodes which are children of the node; and

5 means for generating a node value specification, which is empty if the node has no value;

means for generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

10 means for appending the data buffer to the mXML buffer to form the output document.

26. The system of claim 24, wherein said means for processing said output document in mXML comprises:

15 means for parsing the document, further comprising:

means for parsing a node count representing a count of nodes in the document;

means for parsing a node specification for each of the nodes, further comprising:

means for parsing a node name;

20 means for parsing a child list specifying index values of zero or more nodes which are children of the node;

means for parsing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

means for parsing a node value specification, which is empty if the node has no value; and

means for parsing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value

specifications; and

means for using the parsed document as input for the processing.

27. A method for processing an input document comprising the steps of:

(a) determining whether said input document will be next routed to a target which is capable of processing documents encoded in a machine-oriented extensible markup language ("mXML"); and

(b) converting said input document to an output document encoded in a human friendly extensible markup language ("XML") if said input document is encoded mXML and said target is not capable of processing documents encoded in mXML.

28. The method of claim 27, further comprising the step of:

(c) converting an original document encoded in XML to an input document encoded in a machine-oriented extensible markup language ("mXML"); step (c) being performed before step (a).

29. The method of claim 28, wherein step (a) comprises the steps of:

(a1) identifying a target to which said input document will be next routed;

(a2) determining whether said target is capable of processing documents encoded

in mXML.

30. The method of claim 29, wherein step (a1) comprises parsing said input document.

5 31. The method of claim 31, wherein step (a2) comprises referencing a datastore, said  
datastore storing data identifying a plurality of targets and indicating whether each of said  
plurality of targets is capable of processing documents encoded in mXML.